2009 Fourth International Conference on Internet Monitoring and Protection

Effective Change Detection in Large Repositories of Unsolicited Traffic

Ejaz Ahmed, Andrew Clark, George Mohay
Queensland University of Technology
Brisbane, Australia
{*e.ahmed, a.clark, g.mohay*}*@qut.edu.au*

Abstract

When monitoring unsolicited network traffic automated detection and characterization of abrupt changes in the traffics statistical properties is important. These abrupt changes can either be due to a single or multiple anomalous activities taking place at the same time. The start of a new anomalous activity while another anomalous activity is in operation will result in a new change nested within the previous change. Although detection of abrupt changes to identify malicious activities has received considerable attention in the past, automated detection of nested changes has not been addressed. In this paper a dynamic sliding window cumulative sum (CUSUM) algorithm is proposed to automatically identify these nested changes. The novelty of the proposed technique lies in its ability to automatically detect nested changes, without which interesting activities may go undetected, and its effectiveness in identifying both the start and the end of the individual changes. Using an analysis of real network traces, we show that the identified nested changes were indeed due to distinct malicious behaviours taking place in parallel.

1. Introduction

Today's evolving networks are experiencing a large number of different attacks ranging from system break-ins, infection from automatic attack tools such as worms, viruses, Trojan horses and denial of service (DoS). One important aspect of such attacks is that they are often indiscriminate and target Internet addresses without regard to whether they are bona fide allocated or not. Different tools and techniques have been developed to detect these attacks both on used [1] and unused Internet address spaces [2], [3].

Due to the absence of any live host the traffic observed on unused IP addresses is by definition unsolicited and likely to be either opportunistic or malicious. The analysis of large repositories of such traffic can be used to extract useful information about both on-going and new attack patterns and unveil and unearth unusual attack behaviors. The statistical properties of network traffic are often good candidates for such analysis. The idea is based on the observation that anomalous activities such as denial-of-service attacks, worm outbreak and port-scanning usually results in abrupt changes in statistical properties of the traffic which either remain constant or vary slowly over time during normal traffic operations.

Although significant work has been done in the past to detect abrupt changes with the aim of unveiling both ongoing and new attack patterns, automatic detection of nested changes has not been addressed. We argue that the execution of multiple anomalous activities at the same time will result in nested changes which should be separated to identify the exact cause of different activities. One way to detected multiple attacks is to monitor multiple traffic parameters in parallel such as by monitoring top n destination ports, top n source addresses or top n destination addresses. However parallel attacks such as the one on the same or previously unknown destination port may go undetected using traditional change detection techniques. We believe that multiple anomalous activities will result in multiple changes in traffics statistical properties and effective detection of these multiple nested changes will help in differentiating between nested but distinct anomalous behaviors, without which interesting activities may go undetected. In addition our aim is to keep the technique simple by using minimum information, such as total number of packets per unit time, for automated detection of these parallel activities.

To the best of authors knowledge, this is the first work that uses sliding window based non-parametric cumulative sum method in the context of automated nested change detection from large repositories of unsolicited traffic. The paper is organized as follows. Section 2 reviews related work and outlines our contribution. In Section 3 we formalize the change point detection problem and describe our adaptive sliding window based non-parametric cumulative sum (CUSUM) technique. In Section 4 we discuss the effectiveness of our approach by experimentation on both synthetic and real data sets. Section 5 summarizes the paper and outlines further work.

2. Related Work and Our Contributions

Th objective of change point analysis is to find a point in time where the statistical properties of the observed parameter is transformed in such a way that it no longer remains constant. Due to its simplicity and effectiveness it has been extensively studied by various researchers in dif-

978-0-7695-3612-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIMP.2009.8

ferent problem domains and different techniques have been proposed by these researchers including spectrum analysis, moving average charts, cumulative sum procedure and the Shiryayev-Roberts procedure.

A non-parametric CUSUM algorithm is used by Chan et al.[4] to identify worms which use a hit list of potential target IP addresses to propagate through the network. Each incoming source address is weighted based on heuristics and the total weight in a given time window is calculated. A non-parametric CUSUM is then applied on the total (weighted) source address count which is then subjected to a threshold test using a pre-selected threshold value. Moreover the specific use of the algorithm in detecting only hit list worms makes it less favorable in the analysis of network traffic containing diverse anomalous activities.

Chen et al. [5] have provided a framework to detect distributed denial of service (DDoS) attacks using a distributed change detection algorithm based on the non-parametric CUSUM. In this technique each router is responsible for generating change aggregation trees (CAT) which are then sent to CAT servers for the final decision. The algorithm is tested using an experimental test bed and a synthetic data set.

Yu et al.[6] used a sliding window based technique for anomaly detection in symmetric network traffic. The threshold for change point was calculated dynamically using a sliding window of fixed size. Using this approach on real network traces with synthetically generated SYN flooding attacks, it has been shown that not only the start but also the end of an anomalous event can be detected effectively. However it was observed by the authors in [7] that the use of a fixed size sliding window does not provide satisfactory performance during either normal or transient periods.

The work most closely related to our proposed technique is by Jian et al. [7]. The authors have provided a parameter estimation technique based on a block-wise least square algorithm. The proposed scheme consists of a change detection algorithm and a variable length sliding window. Although the proposed algorithm was effective in detecting both abrupt and gradual changes, our proposed technique is more robust in detecting not only different changes associated with different anomalies but also by being able to detect the start and the end of nested changes. Moreover the authors have used a pre-selected threshold to identify the start and the end of the change point in contrast to the dynamic threshold adjustment technique used in our proposed algorithm.

In this paper, a new adaptive nested change point detection technique is proposed which is extended from the single change point detection scheme reported in [8]. An adaptive method is used to learn the upper bound on the estimated mean (α) in contrast to the previous work where it is selected based on some heuristics or experimental results. The novelty of the proposed technique lies in its ability to automatically detect any number of nested changes and its effectiveness in identifying both the start and the end of the individual changes and is validated through experimentation using both synthetic data and real network traces. In summary, our contributions are highlighted below, the details of which are given in subsequent sections:

1) Use of a sliding window based non-parametric CUSUM algorithm to automatically detect nested change points and their corresponding end points. We argue that identification of nested anomalous activities is important to identify the exact cause of change in traffic dynamics and to effectively distinguish between different sets of anomalous activities, without which interesting activities may go undetected.
2) Use of the proposed technique to analyze large repositories of unwanted traffic such as one observed while monitoring a large number of unused IP addresses. Although the proposed technique can be used to identify anomalous activities embedded in normal network traffic, our aim is to use it in identifying new and unusual attack patterns in large collection of unwanted traffic.
3) Investigating the performance of the proposed technique by using both synthetically generated data sets and by using 18 months of real network traffic collected from a dedicated unused class C address block.

3. Change Point Detection

In a real time network traffic analysis, estimating traffic distribution both before and after a malicious event is rather a difficult task if not impossible due to the lack of a complete model. In change detection, this problem can be solved using a non-parametric CUSUM method as described by Blazek et al. [9]. In [8], the authors adopted a non-parametric CUSUM approach to detect single change in traffic parameters because of its simplicity as no assumption about underlying process probability distribution is required a priori.

In order to describe the change detection algorithm let t_1, t_2,...,t_n be the fixed time instances and N_n be the number of packets received by the monitoring system at time t_n. A malicious activity at time t_k will result in the change in statistical properties of the observed traffic parameter. If μ_k be the mean traffic rate before and $\bar{X}_k$ be the mean traffic rate after the malicious event, then the CUSUM score S_k can be calculated by

$$S_k = max\left\{0, S_{k-1} + N_k - \mu_k - \alpha \cdot \bar{X}_k\right\} \quad (1)$$

where α is a tunning parameter belonging to the interval (0,1) and is considered to be an upper bound on the estimated post-change traffic rate $\bar{X}_k$. The choice of tuning parameter, α, effects the performance of the algorithm as selecting too small or too large an α value will increase the false alarm rate. The α value can either be replaced with a constant

value based on experimentation [8] or can be calculated dynamically [10] using equation given below:

$$0 \leq \alpha \leq 1 - \frac{\mu_k + h_k/T}{\bar{X}_k} \quad (2)$$

where h_k is the detection threshold at time t_k and T is the maximum time in which the change should be detected. The average number of packets at time t_k can be estimated iteratively using an exponential weighted moving average (EWMA):

$$\bar{X}_k \quad = \quad (1-\beta) \cdot \bar{X}_{k-1} + \beta \cdot N_k \quad (3)$$

where $0 < \beta < 1$ is a smoothing factor which gives more weight to the current observation. The CUSUM score, S_k, given in Equation 1 is then subject to a threshold test h in order to detect a change, S_k value greater or equal than the threshold will indicate a change in parameter properties. The threshold value h at time t_k can be calculated as

$$h_k^{start} = \sigma_{k-1}, \quad h_k^{end} = 0.25 \cdot h_k^{start} \quad (4)$$

where h_x^{start} and h_x^{end} are the threshold values for the start and the end of the change point at time t_k and σ_{k-1} is the standard deviation of the data elements in current window at time t_k . In order to reduce the false alarm rate, an additional counter τ is used along with h_k^{end} to mark the end of a change point, the alarm is not canceled until timer τ reaches a specified value, see Equation 5.

$$Alarm = if(S_k \leq h_x^{end}\, AND\, \tau \geq 2, terminate, resume) \quad (5)$$

3.1. Nested Change Points

The change in traffic dynamics can be due to a single anomalous activity or multiple anomalous events taking place in parallel at the same time. The recent work in change point identification detects parallel anomalous activities by monitoring multiple parameters in parallel such as by monitoring top n destination ports or top n source addresses. We argue that the parallel anomalous activities can be detected using minimum information such as total number of packets per unit time as a single change point can be due to different anomalous activities executing in parallel, resulting in a significant change in already changed statistical properties of the parameter under investigation. This will forced us into the investigation of what we call nested change points. Automated identification of these nested changes can help in identifying parallel anomalies which might go undetected using traditional change detection techniques described above.

In order to make our point let us assume that on port x at time t a network experience a malicious activity which continued till t', such that $t' > t$. If at time k a different port y is attacked, given $t' > k > t$, then the dynamics of observed traffic parameter will multi-transform first at time t and then at time k resulting in nested change points. Even the outbreak of different malicious activity at time k on the same port x, given $t' > k > t$, will result in the multi-transformation of parameters statistical properties. Identification of these nested change points is critical not only in effective categorization of different attack behaviors but also to identify exact cause of change in traffic dynamics.

3.2. Nested Change Detection

In order to detect nested change points, we have extended the variable length sliding window based approach proposed in [8]. This can be achieved by keeping track of traffic dynamics before and after each change point. Let W_{t_n} be the sliding window holding data points at time t_n and $W^x_{t_{k-1}} = W_{t_n}$ and $W^x_{t_{k+cons}}$ be the pre-change sliding window holding data elements before change and the fixed length sliding window holding the data points after the change x at time t_k respectively, where $x = 1, 2, ..., n$. Let us assume that the change x is ended at time t_e where $t_e > t_k$. During that time a new anomalous activity at time t_z results in a new change $x+1$, where $t_e > t_z > t_k$, then change $x+1$ and x are considered to be the nested change points.

Under this situation $W^{x+1}_{t_{z-1}} = W^x_{t_{k+cons}}$ will be the pre-change window and $W^{x+1}_{t_{z+cons}}$ will be the fixed length sliding window holding data points after the change $x+1$ at time t_z. Every new data point at time t_{z+1} will first be compared with h^{end}_{x+1} to identify the end of change $x+1$, if the test is successful, and then be compared with h^{end}_x to identify the end of the previous change x. After the termination of all nested changes, any new data point will be analyzed for a possible change in traffic dynamics using dynamic sliding window W_{t_n}. If no change point is identified then the data point is added in the sliding window and the window is slided or expanded based on its current length and the maximum length. Note that the sliding window W_{t_n} will gradually be filled with data points related to normal activity and is being used as a base line behavior to identify changes in traffic statistical properties.

4. Evaluation

In this section the performance of our proposed method is evaluated using datasets consisting of both synthetic and real network traces.

4.1. Experimental Setup

To study the effect of window sizes on the performance of the proposed algorithm, windows of different sizes were used with the synthetic data set. As noted in [8], it is observed that selecting a window size greater than 60 has no or little effect on the performance of the proposed algorithm. Based on the experimental results the maximum size of

window before it starts sliding is selected to be 100 where as the constant size of the sliding window during the change is 5. Due to space limitations the results of this experimentation are not included in the paper. The other parameters including α and h_k were calculated dynamically.

4.2. Synthetic Data

To use the proposed technique for the analysis of large repositories of unsolicited traffic, a synthetic data set which closely approximates such network traces is created. For this, 18 months of real network traffic collected from a dedicated class C darknet is first analyzed and any outliers in the traffic removed. The traffic dynamics including mean and standard deviation of the network traffic is then calculated which are used to generate a synthetic data set consisting of 1000 data points. Using the generated data set as noise, different anomalous events having both high intensity attacks, whose mean amplitude is 250% higher than the mean traffic rate, and low intensity attacks, whose mean amplitude is 50% higher than the mean traffic rate were generated [11]. Table 1 provides a summary of the anomalous events generated in synthetic data set.

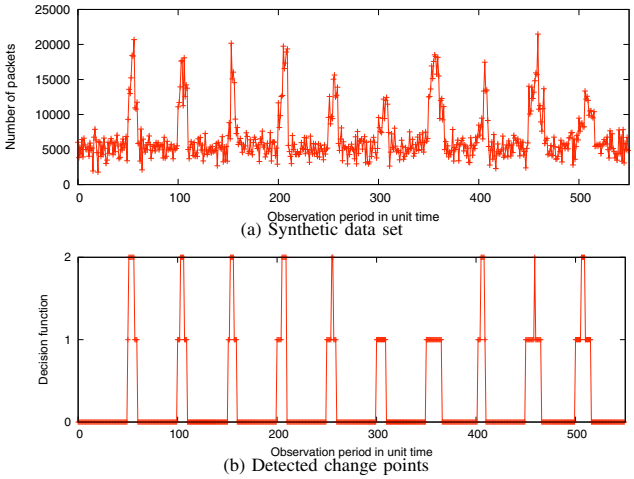
Table 1. Synthetic data

No	Intensity		Duration		Start time		Change ending first
	Data 1	Data 2	Data 1	Data 2	Data 1	Data 2	
1	High	High	10	4	51	54	Change 2
2	High	Low	10	4	101	104	Change 2
3	Low	High	10	4	151	154	Change 2
4	High	High	10	4	201	206	Same
5	High	Low	10	4	251	257	Same
6	Low	Low	10	4	301	306	Same
7	High	High	10	10	351	357	Change 1
8	Low	High	10	4	401	406	Change 2
9	High	High	10	10	451	457	Change 1
10	Low	High	10	10	501	507	Change 1

The data set consists of ten nested (anomalous) changes. The change duration is specified in terms of observation period. For example, the duration of 10 means that the attack is continuous for 10 consecutive observation periods. The last column specifies which change ends first. For example in the first row Change 2 ends before Change 1, and in fourth row both changes ends at the same time.

Figure 1 (a) shows the data set used in the evaluation of the proposed technique. Figure 1 (b) shows the result of applying the proposed technique, where 1 represents Change 1 and 2 represents Change 2. Note that the algorithm was successful in detecting both start and end of nested changes in all cases. However the algorithm mis-identified the end of Change 1 in cases where Change 1 ends before the end of Change 2 (no 7,9 and 10 in Table 1) and Change 2 (no 2,3,4 and 8 in Table 1). The algorithm completely missed Change 2 (6 and 7 in Table 1). Whereas in 1,5,9 and 10 the algorithm mis-labeled either the start or end of Change 2.

It is observed that the effective identification of nested events largely depends upon their relative intensity. In addition the absence of any extra information, about different events, using only the combined traffic will make the algorithm largely dependent on relative intensity of the events and spread of data points within each event. The only information available in this experiment was the total number of packets per unit time which makes it difficult to effectively identify nested events.



(a) Synthetic data set

(b) Detected change points

Figure 1. Synthetic data set

Although the algorithm was successful in identifying the existence of nested events in most cases, 8 out of 10, we believe that the performance of the algorithm can be improved by considering extra information about events such as destination port targeted, number of unique source IP addresses and number of unique destination addresses targeted along with total number of packets and is part of our future work.

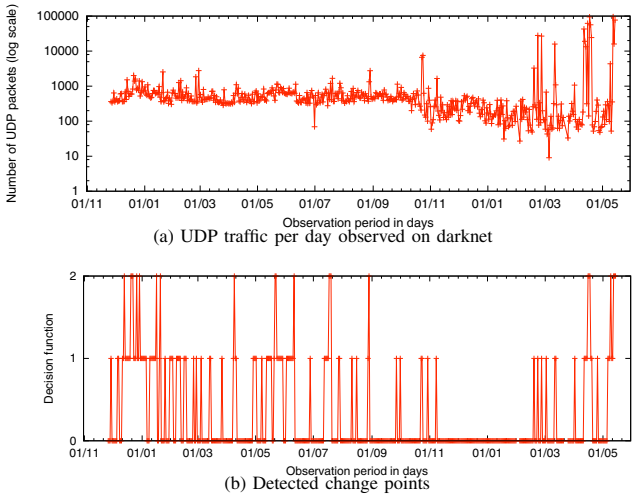
4.3. Real Network Trace

In order to evaluate the applicability and usability of the proposed technique on real data, we have tested the algorithm on 18 months of data collected from dedicated class C darknet. In this section, the results of the proposed technique on UDP traffic collected on the darknet between 27 November 2006 and 15 May 2008 will be discussed.

Figure 2 shows the UDP traffic observed on the darknet during the monitoring period and corresponding change points detected using the proposed technique.

In this section, due to space limitation we will limit our discussion to the three out of ten nested change points identified by our algorithm. The summary of nested change points observed on UDP traffic is given in Table 2. The value in brackets are the percentages of traffic observed on the specific destination port during the change point and the pre-nested columns gives the values during the first change point.

The first and the longest change point was observed on 12/12/2006 which continued for over three weeks. During this period four different nested change points were identified, on 14^{th}, 21^{st}, 27^{th} and 30^{th} of December 2006. The primary cause of this change was due to a sharp increase in the traffic on destination port 137, more that 70% of the total traffic was observed on port 137. This port is used by NETBIOS name service and also by trojan Msinit. During analysis it was observed that all of the traffic on this port was from small number of source IP addresses, 2 to 3 sources per day. Also the source ports used by these sources were either 137 or 1024+n which confirms the existence of worm looking for unprotected network shares as a first step to enumerate and maybe exploit these open file shares.



(a) UDP traffic per day observed on darknet

(b) Detected change points

Figure 2. Darknet traffic trace (UDP)

The nested change point identified on 14^{th} of December 2006 was due to the increase in traffic on two different destination ports 1026 and 1027, more than 75% percent of the total traffic. This is due to the increase in MS Messenger NetSend spamming actvity which is generally related to the pop up messages on MS machines. By default, that "messaging" service runs on UDP/1026 for Windows 2000 and Windows XP, but it can be set to different ports. It is interesting to note that after the end of the first nested change no activity was obserevd on the ports 1026 and 1027.

The second nested change on 21^{st} of December 2006 was due to the increase in traffic on ports 1031, 1032 and 1033. It is interesting to note that the pre-nested destination ports have now two ports, 137 and 1434, collectively receiving more than 74% of the total traffic. The traffic observed on port 1434 was due to MS_SQL Slammer worm that tries to exploit the buffer overflow vulnerability without being authenticated by the server. However in the absence of any extra information such as destination ports, it is not possible for the algorithm to distinguish between the first change point, port 137, and the change point before the second nested change, port 137 and 1434. We believe that this can be improved by monitoring different traffic dynamics in parallel as will be discussed further in the coming section.

The third nested change on 27^{th} of December 2006 was due to increase an in traffic on ports 137 and 1434. In this case our darknet observed a large increase in the number of packets targeting on these two ports, a total of 1415 UDP packets were observed in the third change point in contrast to just 669 packets in the pre-nested change point. The fourth nested change point was again due to the increased activity on ports 1031,1032 and 1033.

During the analysis of the above change point, it was observed that the total number of unique sources during each day of the change point was almost the same, around 75 to 80 unique sources on each day, and only small number of destination ports, less than 15, were targeted during this three weeks period.

The second change point started on 10/01/2007 and ended on 17/01/2007. The second change point was due to the increased MS_SQL slammer activity on port 1434 and increased traffic on port 137, receiving more that 52% and 39% of the total traffic respectively during first change point. The nested change point was observed on the last day of this change point, on 17/01/2007. It was observed that the nested change point was due to the increase in activity on the same two ports. This is due to the fact that during the first change the average number of packets observed per day was 659 which increased to 968 per day during the nested change points. In this case both changes ended on the same day.

Even though the same set of destination ports were targeted during the pre-nested and nested change points, identification of the nested change points in this case will alert to the possibility of changes in other traffic dynamics such as number of unique sources, number of unique payloads, number of destination addresses and even increase in malicious activity from the same source address.

The ninth change point was observed from 12^{th} to 20^{th} of April 2008. The nested change was observed on 16^{th} which continued for three days. Note that on each day of both changes different sets of destination ports were targeted, with no port targeted on multiple days, and more than 98% of the traffic targeted the same destination IP x.x.x.221. It is important to note that more than 98% of the source addresses related to this activity were observed on the darknet only on the day of the activity. During this event an increase in unique sources is also observed.

Although we do not know at this stage what really caused this kind of behavior, it is the main cause of a huge spike in the number of UDP packets collected by the darknet in the respective days. The close proximity of these relative activities and their distinct behavior is indeed due to some unknown but interesting phenomenon which we aim to analyze in detail and is part of our future work. Moreover as the traffic collected on the darknet is by definition malicious, it is very difficult to comment on the false alarm rate of the

Table 2. Summary of nested change points observed in a UDP traffic

CP	Date		Average Packets		Average Sources		Destination ports	
	First CP	Nested CP	Pre-Nested	Nested	Pre-Nested	Nested	Pre-Nested	Nested
1	12/12-06/01/07	14/12/2006	550	1515	106	73	137(70.5)	1026-1027(75.9),137(16.7)
		21-23/12/06	618	1667	79	71	137(51.8),1434(25.5)	1031-1033(39.1),1434(18.4)
		27/12/06	669	1415	78	81	137(45.2),1434(29.6)	137(72.1),1434(13.4)
		30/12/06	657	1338	78	75	137(44.9),1434(31.9)	1031-1033(56.57),1434(19.5),137(18.9)
2	10-17/01/07	17/01/07	659	968	71	81	1434(52.5),137(39.1)	1434(67.8),137(26.2)
3	20-21/01/07	21/01/07	651	2543	63	54	137(39.4),1434(33.3)	1030-1035(60.4),137(10.3),1434(7.1)
4	08-11/04/07	09/04/07	628	1101	67	83	1026(46.5),137(41.4)	1026(56.7),137(23.2)
5	21-30/05/07	22-23/05/07	778	1133	106	104	137(67.8),1434(18.4)	137(56.8),33667(25.9),1434(7.1)
6	03-11/06/07	11/06/07	668	1135	62	75	137(82.5)	137(87.7)
7	13-20/07/07	18-20/07/07	555	1087	61	74	137(49.5),1030-1034(17.3)	137(43.1),1030-1034(34)
8	28-29/08/07	29/08/07	1332	2749	317	811	137(30.1),1026-1028(57.65)	137(9.34),1026-1028(83.8)
9	12-20/04/08	16-18/04/08	18757	52257	276	844	13398(56.2),13048(22),13408(17.2)	13763(30.7),13979(5.5),13493(58.3)
10	07-15/05/08	10/05/08	244	4342	49	357	137(35.3),53(35)	13284(97.6)
		13-15/05/08	229	63021	56	1166	137(22.7),13593(21.4)	13274(50),62997(20.7),13205(14.2)

proposed technique. On the other hand our experiment with the proposed technique shows that all the nested changes identified by the proposed technique was indeed distinct malicious behaviors observed by the darknet. While the proposed algorithm can be used in parallel, for example monitoring top 10 destination ports in parallel, our aim is keep the technique simple by using minimum information, such as total number of packets per unit time, for automatic detection of these parallel activities.

5. Conclusion and Future Directions

In this paper, we have proposed an adaptive sliding window based technique for automatically detecting nested changes. We extended the sliding window technique [8] to detect any number of nested changes. The motivation behind our work was to detect nested but distinct anomalous behaviors in large repositories of unsolicited traffic using minimum information. The effectiveness of the proposed technique is analysed using both synthetic and real network traces. During experimentation with synthetic data it is noted that the proposed technique effectively identified nested events in most cases. The applicability and usability of the proposed algorithm is analysed with the help of real network traces. It is observed that the proposed technique effectively identified nested behaviors in UDP traffic collected from the darknet. Even though in both synthetic and real network traces at most two nested changes were identified, we believe that the algorithm can detect any number of nested changes.

We suggest that the results can be improved by parallel analysis of multiple traffic parameters. Extending the proposed approach to multiple traffic parameters and correlation of detected change points may help in identifying the degree of involvement of these parameters in the detected change.

References

[1] G. White, E. Fisch, and U. Pooch, *Computer System and Network Security*. CRC Press, 1995.

[2] X. Jiang and D. Xu, "Collapsar: a vm-based architecture for network attack detention center," in *Proceedings of the 13th conference on USENIX Security Symposium (SSYM)*. Berkeley, CA, USA: USENIX Association, 2004, pp. 2–2.

[3] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, and S. Sinha, "Practical darknet measurement," *40th Annual Conference onInformation Sciences and Systems*, pp. 1496–1501, March 2006.

[4] J. Chan, C. Leckie, and T. Peng, "Hitlist worm detection using source IP address history," *Proceedings of Australian Telecommunication Networks and Applications Conference*, 2006.

[5] Y. Chen, K. Hwang, and W.-S. Ku, "Collaborative detection of DDoS attacks over multiple network domains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1649–1662, Dec. 2007.

[6] M. Yu and X.-Y. Zhou, "An adaptive method for anomaly detection in symmetric network traffic," *Computers & Security*, vol. 26, no. 6, pp. 427–433, 2007.

[7] J. Jiang and Y. Zhang, "A novel variable-length sliding window blockwise least-squares algorithm for on-line estimation of time-varying parameters," *International journal of adaptive control and signal processing*, vol. 18, no. 6, pp. 505–521, 2004.

[8] E. Ahmed, A. Clark, and G. Mohay, "A novel sliding window based change detection algorithm for asymmetric traffic," *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*, pp. 168–175, Oct. 2008.

[9] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods," in *IEEE Workshop on Information Assuarance and Security*, 2001.

[10] H. Kim, B. Rozovskii, and A. Tartakovsky, "A nonparametric multichart cusum test for rapid detection of DoS attacks in computer networks," *International Journal of Computer and Information Sciences*, vol. 2, no. 3, pp. 149–158, 2004.

[11] V. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks," *Computer Communications*, vol. 29, no. 9, pp. 1433–1442, 2006.